

### REMARKS

|                                |   |
|--------------------------------|---|
| Claims previously in issue:    | 1-27  |
| Claims amended:                | 1, 2, 4, 5, 10, 11, 13, 14, 19, 20, 22, 23          |
| Claims cancelled:              | 3, 12, 21   |
| Claims added:                  | None  |
| Claims rejected                | 1-3, 8-12, 17-21, 26-27                             |
| Claims indicated as allowable: | 4-7, 13-16, 22-25, if rewritten in independent form |
| Claims remaining in issue:     | 1-2, 8-11, 17-20, 26-27                             |

### **The Invention**

The invention includes a system, method, and program for parallelizing applications of certain script-driven software tools. In the illustrated embodiment, scripts in the software tool scripting language are *automatically* analyzed in order to produce a *specification* for a parallel computation which is functionally equivalent to the original script. This is accomplished by (a) parsing the script into statements; (b) constructing a serial dataflow graph from the parsed statements; and (c) constructing a parallel dataflow graph from the serial dataflow graph.

The parallel computation specification is then executed by a parallel runtime system, which causes multiple instances of the original software tool and/or supplemental programs to be run as parallel processes. The resulting processes will read input data and produce output data, performing the same computation as was specified by the original script. The combination of the analyzer, runtime system, original software tool, and supplemental programs will, for a given script and input data, produce the same output data as the original software tool alone, but has the capability of using multiple processors in parallel for substantial improvements in overall "throughput".

### **The §103 Rejection**

The Examiner has again rejected claims 1-3, 8-12, 17-21, 26-27 under 35 U.S.C. §103(a) as being obvious, this time in view of Iwasawa *et al.* 5,151,991. Applicant respectfully traverses this rejection with respect to the claims in issue.

Iwasawa teaches a method and system for recompiling serial code in order to parallelize inner and outer loops. See Iwasawa, col. 1, l. 50 to col. 2, l. 13. The loops are taught as repetitive

executions of the same statements with possibly differing parameters at each execution (*e.g.*, DO loops).

Iwasawa is thus narrowly focused on a particular type of parallelization. Contrary to the Examiner's assertions, Iwasawa teaches nothing about generation or construction of *dataflow graphs*. Dataflow graphs comprise a set of vertices coupled by directed edges, or links (see, *e.g.*, Fig. 3 of the application). Iwasawa detects loops and assigns particular loops to particular processors. Nothing in Iwasawa teaches or suggests a way of solving the completely different problem of parsing a script into statements, constructing a serial dataflow graph from the parsed statements, and constructing a parallel dataflow graph from the serial dataflow graph.

The claims have been amended to clarify that they are directed to methods, systems, and programs for parallelizing a computer application program based on a script of a script-driven software tool by automatically analyzing the script and parsing the script into statements, constructing a serial dataflow graph from the parsed statements, and constructing a parallel dataflow graph from the serial dataflow graph.